

Esame di Calcolatori Elettronici T

27 Gennaio 2016 (Ing. Informatica)

Esercizio 1

In un sistema basato sul processore DLX con **1 GB di EPROM** mappata negli indirizzi bassi e **128 MB di RAM** mappata negli indirizzi alti sono presenti due gruppi di periferiche A e B alle quali il DLX è *sensibile* in modo mutuamente esclusivo. Ciascun gruppo di periferiche A e B è formato da 4 dispositivi identici a 8 bit già progettati e denominati, rispettivamente, **INPUT_PORT_A_i (i=0,1,2,3)** e **INPUT_PORT_B_i (con i=0,1,2,3)**. Tutte le 8 periferiche sono in grado di ricevere dati dall'esterno da altrettante unità esterne utilizzando il protocollo di *handshake*. **All'avvio**, dovranno essere connesse al DLX le 4 periferiche appartenenti al **gruppo A**.

Allorché le periferiche del gruppo A sono tutte in grado di fornire un dato al DLX, il processore deve leggere con un unico trasferimento i 4 byte dalle 4 periferiche. Inoltre, se al termine di tale trasferimento risultasse che la somma dei 4 byte letti è un numero *pari* (verifica eseguita da una opportuna rete logica), è necessario rendere sensibili al DLX le periferiche del gruppo B e ignorare quelle del gruppo A. In caso contrario (i.e., *dispari*) debbono rimanere connesse al sistema le periferiche del gruppo A. Quando sono connesse le periferiche del gruppo B deve essere mantenuta la stessa modalità di lettura e condizione di alternanza: definita in precedenza. Ovvero, quando verificata la condizione di alternanza (i.e., *pari*), disconnessione del gruppo B e connessione del gruppo A.

In entrambi i casi, **al termine di ogni lettura dalle periferiche attive**, i 32 bit letti debbono essere trasferiti in memoria **via software**, all'indirizzo FFFF0000h se in quel momento è attivo il gruppo A e all'indirizzo FFFF0004h in caso contrario. A tal proposito si escogiti una strategia che consenta di **velocizzare l'intera procedura software** e allo stesso tempo di **non utilizzare istruzioni di branch/set all'interno dell'interrupt handler**.

- Evidenziare e risolvere eventuali criticità
- Indicare i segnali di chip select di tutte le periferiche, memorie e di eventuali altri segnali presenti nel sistema progettato
- Progettare, **ottimizzando le risorse utilizzate**, tutte le reti necessarie al funzionamento del sistema indicando chiaramente le connessioni di tutte le periferiche e le memorie ai bus del DLX
- Scrivere il codice dell'*interrupt handler* (**senza utilizzare istruzioni di branch/set**) che gestisce i trasferimenti. Si assuma che i registri da R20 a R30 possano essere utilizzati senza la necessità di dover preservare il loro contenuto

Esercizio 2

Spiegare cosa è la *metastabilità* e quali metodologie possono essere adottate per evitarne l'origine.

Esercizio 3

Spiegare perchè le ~~alee~~ di dato dovute alle istruzioni di LOAD sono critiche e come sono gestite nel DLX pipelined.

MAPPING

EPROM 1 GB: $0 \times 0000\ 0000 \rightarrow 0 \times 3fff\ ffff$ ($4 \times 256\text{ MB}$)

RAM 128 MB: $0 \times f800\ 0000 \rightarrow 0 \times ffff\ ffff$

CS_EPROM_0 = $\overline{BA31}\ BA30\ \overline{BE0}$

CS_RAM_0 = $BA31\ BE0$

CS_EPROM_1 = $\overline{BA31}\ BA30\ BE1$

CS_RAM_1 = $BA31\ BE1$

CS_EPROM_2 = $\overline{BA31}\ BA30\ BE2$

CS_RAM_2 = $BA31\ BE2$

CS_EPROM_3 = $\overline{BA31}\ BA30\ BE3$

CS_RAM_3 = $BA31\ BE3$

CS_INPUT_0 = $\overline{BA31}\ BA30\ \overline{BA2}\ BE0$ $0 \times 4000\ 0000$

CS_INPUT_1 = $\overline{BA31}\ BA30\ \overline{BA2}\ BE1$ $0 \times 4000\ 0001$

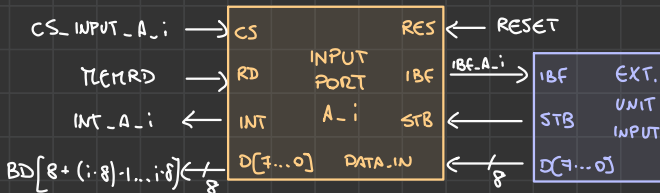
CS_INPUT_2 = $\overline{BA31}\ BA30\ \overline{BA2}\ BE2$ $0 \times 4000\ 0002$

CS_INPUT_3 = $\overline{BA31}\ BA30\ \overline{BA2}\ BE3$ $0 \times 4000\ 0003$

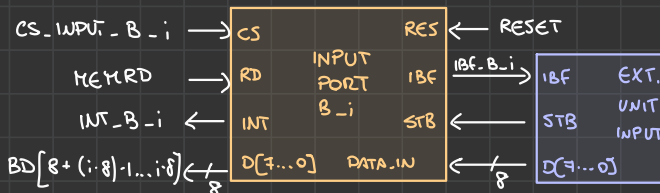
CS_READ_STATUS = $\overline{BA31}\ BA30\ \overline{BA2}\ BE0$.MEMRD $0 \times 4000\ 0004$

FORTE I/O

GRUPPO A



GRUPPO B

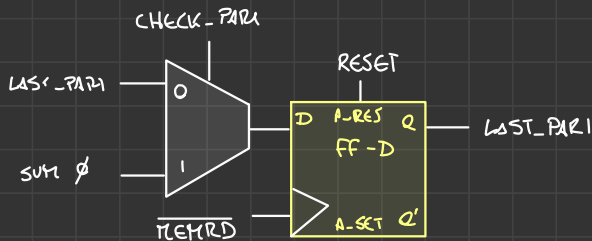
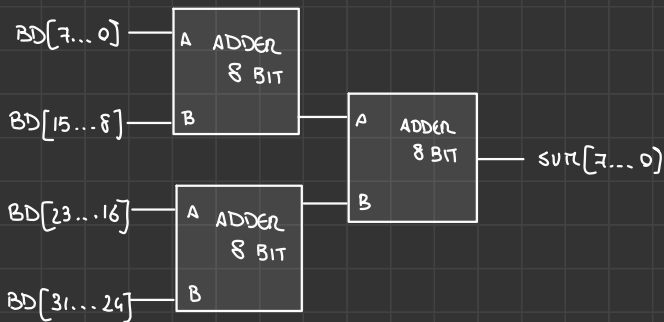


$$CS_INPUT_A_i = CS_INPUT_i \cdot \overline{LAST_PAR1} \cdot IBF_A_i$$

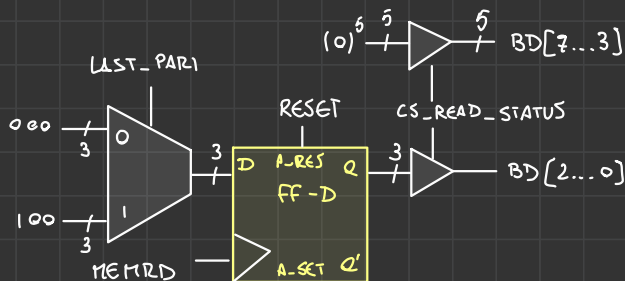
$$CS_INPUT_B_i = CS_INPUT_i \cdot \overline{LAST_PAR1} \cdot IBF_B_i$$

$$INT(TO_DLX) = (INT_A_0 \cdot INT_A_1 \cdot INT_A_2 \cdot INT_A_3) \cdot \overline{LAST_PAR1} \cdot \\ + (INT_B_0 \cdot INT_B_1 \cdot INT_B_2 \cdot INT_B_3) \cdot \overline{LAST_PAR1}$$

Rete per la somma dei dati trasferiti



$$CHECK_PAR1 = CS_INPUT_A_0 \cdot CS_INPUT_A_1 \cdot CS_INPUT_A_2 \cdot CS_INPUT_A_3 + CS_INPUT_B_0 \cdot CS_INPUT_B_1 \cdot CS_INPUT_B_2 \cdot CS_INPUT_B_3$$



CODICE

0h	LHI	R20, 0x4000
4h	LBV	R21, 0x0004(R20)
8h	LHI	R22, 0xFFFF
Ch	ADD	R23, R22, R21
10h	LW	R24, 0x0000(R20)
14h	SW	R24, 0x0000(R23)
18h	REG;	

Calcolatori Elettronici T
Ing. Informatica

Traccia soluzione
27 Gennaio 2016

Esercizio 1

Dispositivi e segnali presenti nel sistema:

RAM_128	mappata da F8000000h:FFFFFFFFh, 4 banchi da 32 MB
EPROM_1GB	mappata da 00000000h:3FFFFFFFFh, 4 banchi da 256 MB
CS_INPUT	mappato a 80000000h (lettura dalle 4 periferiche attive)
CS_READ_ATTIVO	mappato a 80000004h (lettura stato segnale ATTIVO_A)

CS_RAM_128_0	= BA31 · BA30 · BE0
CS_RAM_128_1	= BA31 · BA30 · BE1
CS_RAM_128_2	= BA31 · BA30 · BE2
CS_RAM_128_3	= BA31 · BA30 · BE3

CS_EPROM_1GB_0	= BA31* · BE0
CS_EPROM_1GB_1	= BA31* · BE1
CS_EPROM_1GB_2	= BA31* · BE2
CS_EPROM_1GB_3	= BA31* · BE3

CS_INPUT	= BA31 · BA30* · BA2* · BE0 · BE1 · BE2 · BE3	(Vedi nota*)
CS_READ_ATTIVO	= BA31 · BA30* · BA2 · MEMRD	

Nota*: Il segnale **CS_INPUT** inviato a ciascuna periferica sarà ulteriormente condizionato da un segnale che codifica quale dei due gruppi è attivo (**ATTIVO_A**). Inoltre, il segnale **CS_INPUT** è stato condizionato dai quattro segnali BE0, BE1, BE2, BE3 al fine di non abilitare le reti che dipendono da **CS_INPUT** in caso di accessi (letture) non a 32 bit. Con la stessa finalità si condizioneranno i segnali di CS di ciascuna periferica con i segnali IBF come mostrato nelle pagine successive. Si osserva tuttavia che queste precauzioni non eliminano completamente problemi causati da accessi inappropriati alle periferiche. Ad esempio, cosa si verifica se una istruzione accede alle periferiche attive leggendo un byte o una half-word oppure quando non sono asseriti tutti i segnali di interrupt delle periferiche attive?

Nel sistema sono presenti due gruppi di periferiche, denominati A e B, ciascuno composta da 4 periferiche che sono in grado di comunicare con dei dispositivi esterni mediante il protocollo di *handshake*. I 2 gruppi A e B sono alternativamente connessi al DLX.

Il segnale **ATTIVO_A**, ottenuto da una rete logica mostrata nella pagine successive, codificherà quale gruppo di periferiche è in un determinato istante connesso al DLX (gruppo A se **ATTIVO_A** è asserito, gruppo B in caso contrario).

Allorché tutte le periferiche attive in un determinato istante sono in grado di trasferire un dato, deve essere eseguita un'unica lettura a 32 bit dalle 4 periferiche mappate su 4 distinti bus del DLX.

Se, al termine del trasferimento contemporaneo dei quattro byte, la somma degli stessi risultasse pari debbono essere attivate le quattro periferiche dell'altro gruppo e così via.

Si osservi che non è necessario calcolare la somma dei quattro byte ma è sufficiente determinare se la somma è pari. Questa verifica può essere effettuata da una rete logica che analizza solo i bit meno significati dei 4 byte letti dalle periferiche attive.

Infine, una volta letto il dato a 32 bit, deve essere effettuato senza eseguire alcuna istruzione di *BRANCH* o *SET* nell'*interrupt handler*, la scrittura della word letta dalle 4 periferiche attive all'indirizzo **FFFF0000h**, nel caso sia asserito il segnale **ATTIVO_A**, o all'indirizzo **FFFF0004h**, nel caso sia asserito il segnale **ATTIVO_A***.

I due gruppi A e B saranno attivi alternativamente (connessi al DLX in modo mutuamente esclusivo) in funzione del segnale **ATTIVO_A**. Le quattro periferiche di ciascun gruppo sono mappate su 4 bus distinti al fine di poter eseguire una singola lettura a 32 bit.

All'interno di ciascun gruppo (A o B) ogni periferica è codificata con un indice $i = 0, 1, 2, 3$. Pertanto, i CS dei due gruppi di periferiche A e B risultano:

$$\begin{aligned} \text{CS_INPUT_PORT_A} &= \text{CS_INPUT} \cdot \text{IBF_A_3} \cdot \text{IBF_A_2} \cdot \text{IBF_A_1} \cdot \text{IBF_A_0} \cdot \text{ATTIVO_A} \\ \text{CS_INPUT_PORT_B} &= \text{CS_INPUT} \cdot \text{IBF_B_3} \cdot \text{IBF_B_2} \cdot \text{IBF_B_1} \cdot \text{IBF_B_0} \cdot \text{ATTIVO_A}^* \end{aligned}$$

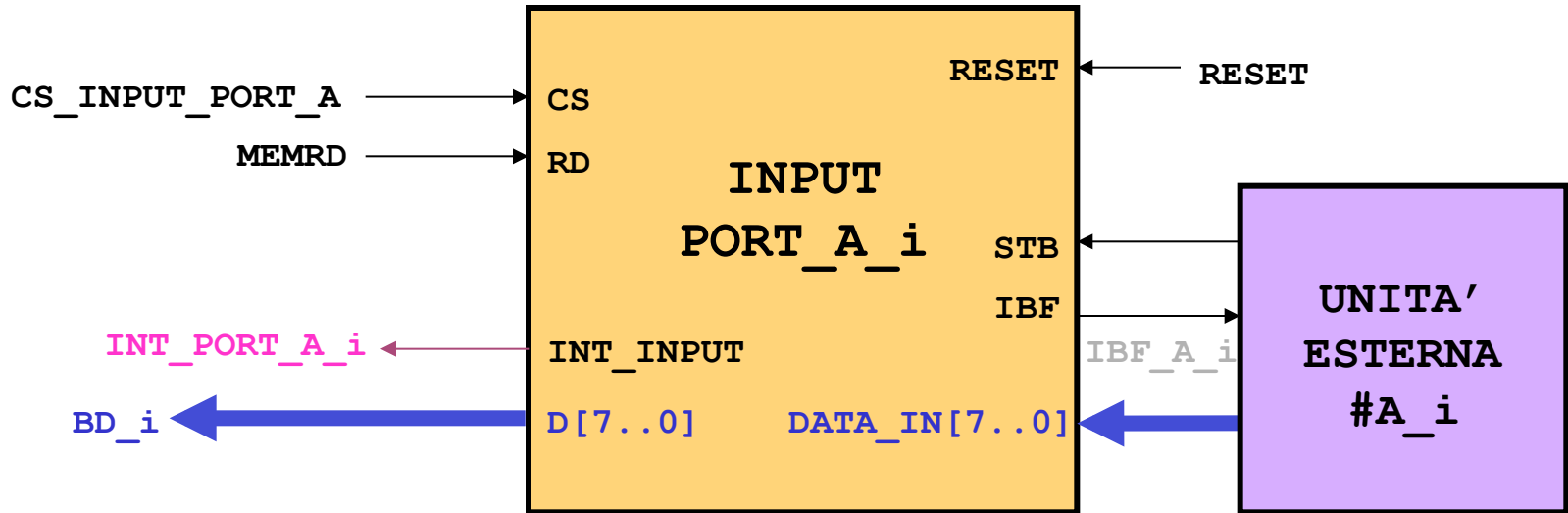
Per semplificare la notazione, nello schema della pagina successiva i 4 bus dati del DLX sono identificati con la notazione **BD_i** seguente:

$$\begin{aligned} \text{BD}_0 &\leftrightarrow \text{BD}[7..0] \\ \text{BD}_1 &\leftrightarrow \text{BD}[15..8] \\ \text{BD}_2 &\leftrightarrow \text{BD}[23..16] \\ \text{BD}_3 &\leftrightarrow \text{BD}[31..24] \end{aligned}$$

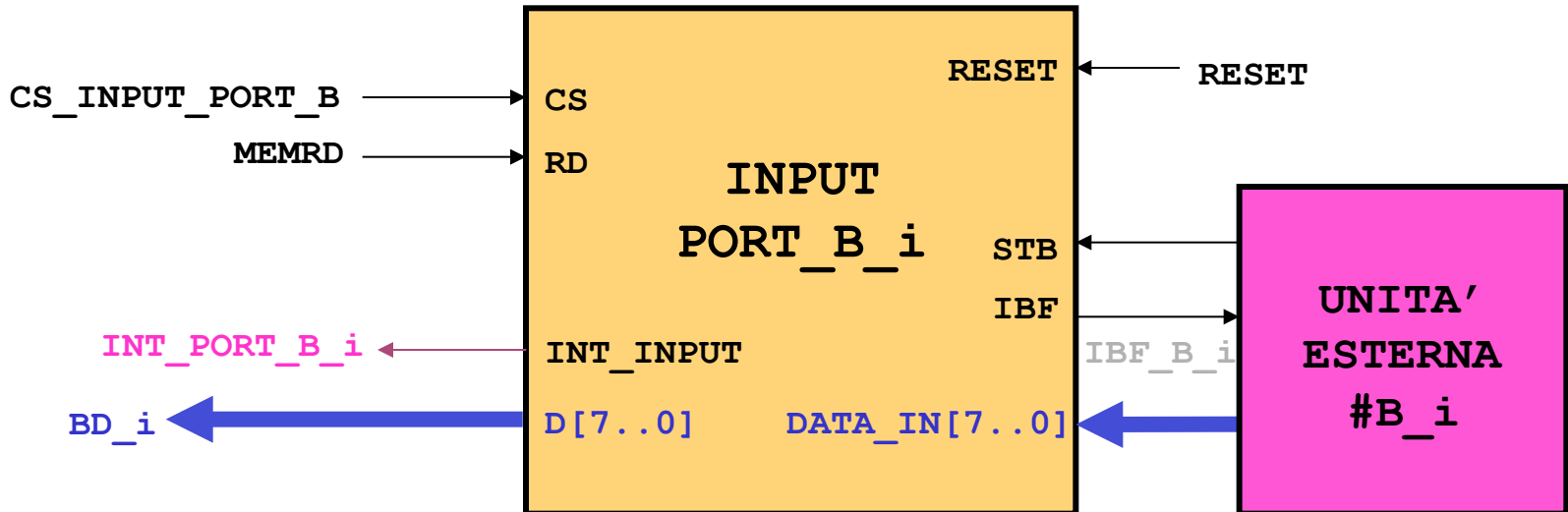
I trasferimenti dal gruppo attivo (codificato dal segnale **ATTIVO_A**) sono possibili solo quando tutte le 4 richieste di interrupt di quel gruppo sono asserite. Pertanto, il segnale **INT_DLX**, inviato al DLX, risulta:

$$\begin{aligned} \text{INT_DLX} &= \text{INT_PORT_A_3} \cdot \text{INT_PORT_A_2} \cdot \text{INT_PORT_A_1} \cdot \text{INT_PORT_A_0} \cdot \text{ATTIVO_A} + \\ &\quad \text{INT_PORT_B_3} \cdot \text{INT_PORT_B_2} \cdot \text{INT_PORT_B_1} \cdot \text{INT_PORT_B_0} \cdot \text{ATTIVO_A}^* \end{aligned}$$

$i = 0, 1, 2, 3$



$i = 0, 1, 2, 3$



Per determinare, nel più rapido tempo possibile al fine di non rendere critiche le operazioni che debbono essere svolte durante il ciclo di lettura dalle 4 periferiche attive, è possibile esaminare solo i bit meno significativi (**BD24**, **BD16**, **BD8** e **BD0**) dei 4 byte letti dalle 4 periferiche.

Se il numero di valori dispari in ciascun byte è in numero dispari la somma sarà dispari, pari in caso contrario.

	BD24	BD16	BD8	BD0		
0	0	0	0	0	-> PARI	1
1	0	0	0	1	-> DISPARI	0
2	0	0	1	0	-> DISPARI	0
3	0	0	1	1	-> PARI	1
4	0	1	0	0	-> DISPARI	0
5	0	1	0	1	-> PARI	1
6	0	1	1	0	-> PARI	1
7	0	1	1	1	-> DISPARI	0
8	1	0	0	0	-> DISPARI	0
9	1	0	0	1	-> PARI	1
10	1	0	1	0	-> PARI	1
11	1	0	1	1	-> DISPARI	0
12	1	1	0	0	-> PARI	1
13	1	1	0	1	-> DISPARI	0
14	1	1	1	0	-> DISPARI	0
15	1	1	1	1	-> PARI	1

La strategia basata sull'analisi dei bit meno significativi dei 4 byte non solo consente di velocizzare la determinazione della parità ma consente anche di risparmiare risorse hardware.

La tabella mostrata nella pagina precedente può essere sintetizzata attraverso l'espressione che segue, che determina la condizione di parità dall'analisi dei 4 bit **BD24**, **BD16**, **BD8**, **BD0** concatenati:

$$\text{SOMMA_PARI} = \text{DEC}(0) + \text{DEC}(3) + \text{DEC}(5) + \text{DEC}(6) + \text{DEC}(9) + \text{DEC}(10) + \text{DEC}(12) + \text{DEC}(15)$$

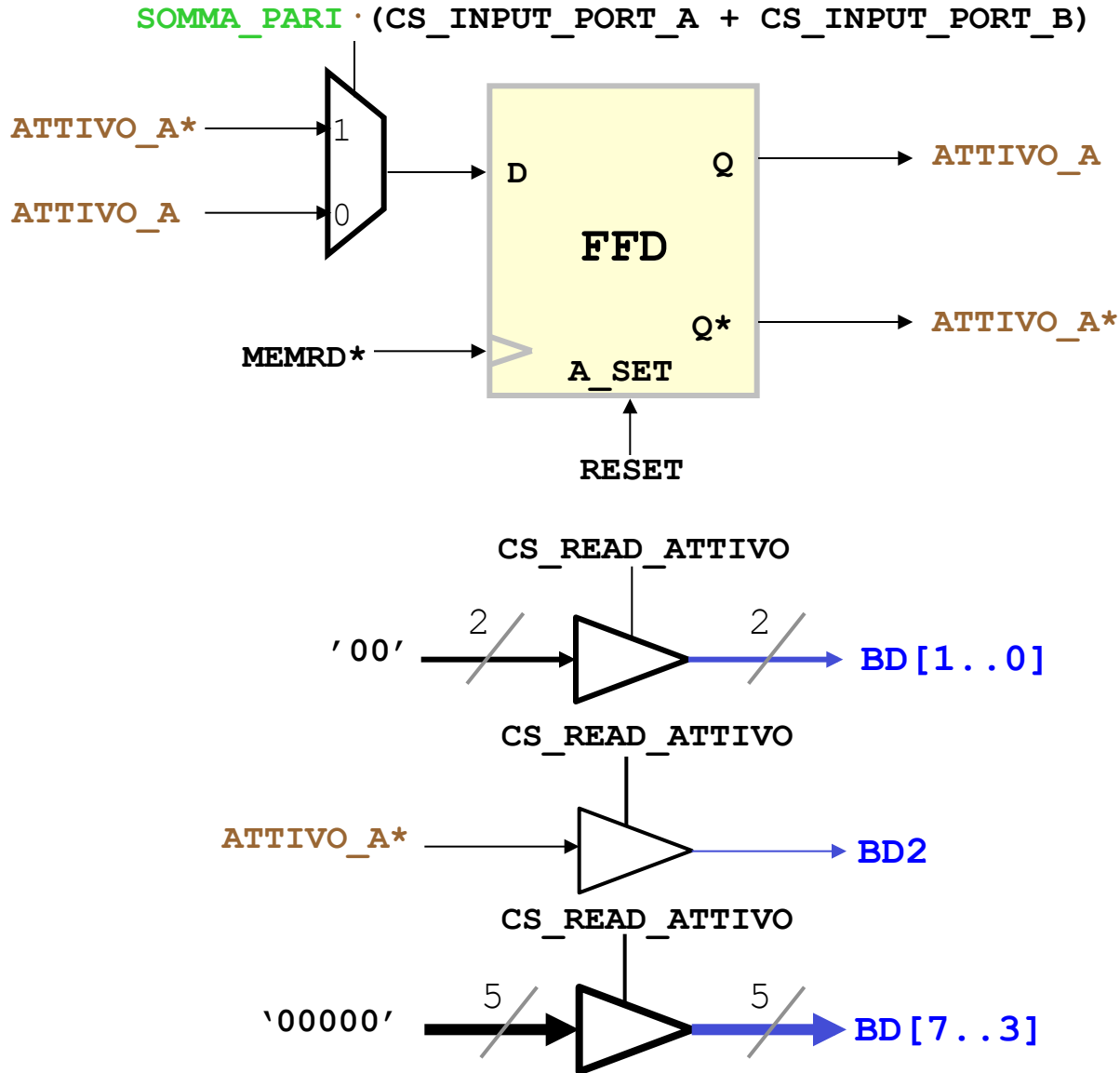
La decodifica dei segnali **BD24##BD16##BD8##BD0** può essere ottenuta mediante un decoder 4:16.

In alternativa, utilizzando meno risorse, è possibile determinare il valore di **SOMMA_PARI*** mediante una rete basata su due livelli XOR (e un NOT per ottenere **SOMMA_PARI**), nel modo seguente:

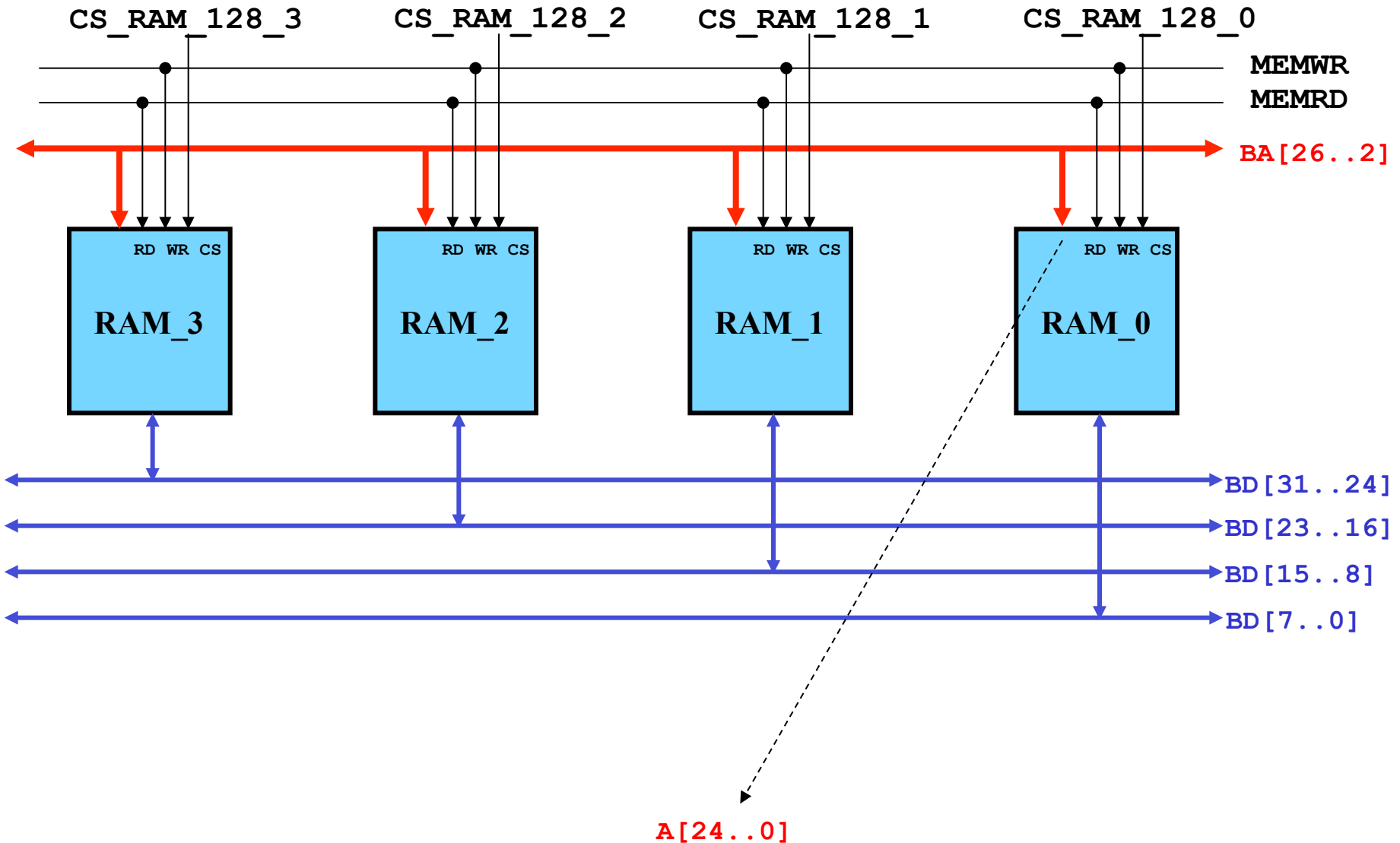
$$\text{SOMMA_PARI*} = (\text{BD24 XOR BD16}) \text{ XOR } (\text{BD8 XOR BD0})$$

Ovviamente, per ottenere il segnale **SOMMA_PARI** riducendo il numero di livelli a 2, è possibile sostituire nel secondo livello di elaborazione lo XOR con uno XNOR (e non utilizzare il NOT).

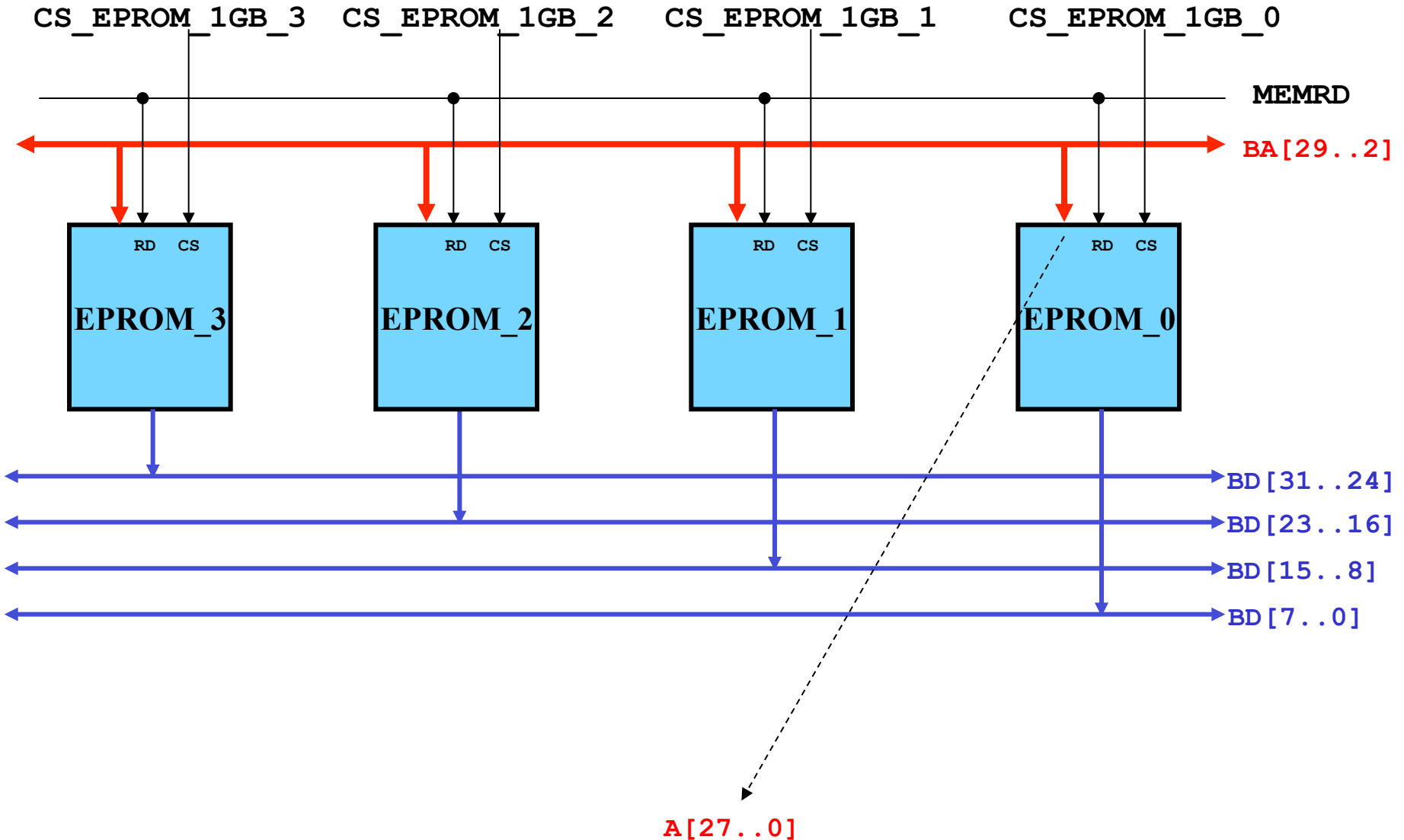
Il segnale **ATTIVO_A** è generato dalla rete seguente. All'avvio il segnale **ATTIVO_A** è asserito. Il segnale è letto dal DLX mediante la rete mostrata in basso (opportunamente elaborato in base alle finalità richieste).



Interfacciamento RAM_128



Interfacciamento EPROM_1GB



Esercizio 2

Rif. lucidi/lezioni.

Esercizio 3

Rif. lucidi/lezioni.